JOURNAL OF COMPLEXITY **18**, 589–611 (2002)
doi:10.1006/jcom.2001.0634

# Verifying Topological Indices for Higher-Order Rank Deficiencies

R. Baker Kearfott and Jianwei Dian

*Department of Mathematics, University of Louisiana at Lafayette, Lafayette, Louisiana 70504*
E-mail: rbk@louisiana.edu

Received November 14, 2000; revised June 24, 2001; accepted September 10, 2001;
published online April 23, 2002

It has been known how to use computational fixed point theorems to verify existence and uniqueness of a true solution to a nonlinear system of equations within a small region about an approximate solution. This can, be done in $\mathcal{O}(n^3)$ operations, where $n$ is the number of equations and unknowns. However, these standard tech-



setting, and in the real setting for odd multiplicity, when the rank defect of the Jacobi matrix at an isolated solution is 1. Here, after reviewing work to date, we discuss the case of higher rank defect. In particular, it appears that $p$-dimensional searches are required if the rank defect is $p$, and that the work increases exponentially in $p$. © 2002 Elsevier Science (USA)

## 1. BACKGROUND

Given a system of nonlinear equations, numerical methods can typically produce an approximation $\check{x}$ to a solution $x^*$. It is then sometimes desirable to compute bounds

$$\begin{aligned}\mathbf{x} &= (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)\\ &= ([\underline{x}_1, \bar{x}_1], [\underline{x}_2, \bar{x}_2], \ldots, [\underline{x}_n, \bar{x}_n],\end{aligned}$$

such that $\check{x}$ is the center of $x$. Specifically, we examine the problem

$$\begin{aligned}&\text{Given } f\colon x \to \mathbb{R}^n \text{ and } \mathbf{x} \in \mathbb{IR}^n, \ \textit{rigorously} \text{ verify:}\\ &\quad\bullet \text{ there exists a unique } x^* \in \mathbf{x} \text{ such that } F(x^*) = 0.\end{aligned} \tag{1}$$



0885-064X/02 $35.00

Here, $\mathbb{IR}^n$ represents the set of $n$-dimensional vectors, as $\mathbf{x}$, whose components are intervals. Also, "rigorously verify' means "use the computer to automatically prove, with the same philosophical validity as a traditional mathematical proof, that a unique solution exists within $\mathbf{x}$." To do this rigorous verification, we construct algorithms that compute, with a finite number of arithmetic operations (including function and derivative evaluations) "yes, there is a unique solution," for all functions satisfying certain properties and boxes $\mathbf{x}$ satisfying certain properties. (For the properties, see Section 1.2 below.) Even though floating point arithmetic is used in these methods, there is no uncertainty in the result due to rounding errors.

To explain Problem 1 more thoroughly, we first introduce some notation, then briefly explain how floating point computations have been used in the past to rigorously verify existence and uniqueness.

### 1.1. *Fundamentals and Notation*

#### 1.1.1. *Interval Arithmetic*

Interval arithmetic, a basic tool in these studies, has been introduced in various works. We assume familiarity with the fundamentals of interval arithmetic. For a relatively succinct but wide-ranging survey, see [14]. Longer introductions include [2, 10, 15, 20, 21]. Thousands of research articles (such as those in the bibliography [3]) have been published, an intrinsic interval data type is fully supported in Sun's Fortran 95 compiler [26], and numerous portable packages are available to supply interval data types in other programming languages. See [18] for information about these, as well as descriptions of succesfull applications, address information for researchers, etc. We briefly sketch the most essential ideas here.

The basic datum in interval arithmetic is the interval, that we denote with boldface, such as $\mathbf{x}$. The most common representation of intervals is in terms of their lower and upper end points; we denote the lower and upper end points of an interval $\mathbf{x}$ by $\underline{x}$ and $\bar{x}$, respectively, that is, $\mathbf{x} = [\underline{x}, \bar{x}]$. We think of intervals as representing values that are not known precisely, but that are known only to lie between the lower bound and upper bound. Operations on intervals are defined as the set of all possible results:

$$\mathbf{x}\ op\ \mathbf{y} = \{x\ op\ y \mid x \in \mathbf{x} \text{ and } y \in \mathbf{y}\}.$$

That is, the result of an interval basic operation is the range of the operation over its arguments. The power of interval arithmetic lies partially in the fact that such ranges can be computed operationally. For instance, for addition,

$$\mathbf{x} + \mathbf{y} = [\underline{x} + \underline{y}, \bar{x} + \bar{y}].$$

(Interval subtraction can be computed similarly; multiplication and division can also be computed operationally, with somewhat more involved formulas.) If operations are composed, an interval evaluation of the resulting expression in general is not equal to the range, but merely contains the range of the resulting expression; moreover, different expressions that are equivalent in real arithmetic give different bounds on the range. For instance, evaluating $f(x) = x - x$ over $\mathbf{x} = [1, 2]$ gives $[1, 2] - [1, 2] = [-1, 1]$. (The overestimation occurs because it is implicitly assumed when substituting $[1, 2]$ for $x$ that the quantity in the first occurrence of $[1, 2]$ is unrelated to the quantity in the second occurrence of $[1, 2]$.) However, regardless of such overestimation, the size of the overestimation of the range tends to zero as the widths of the intervals representing the independent variables tends to zero; for a precise statement of this, see [15, Sect. 1.1.7] and the references therein.

Rigor in interval arithmetic comes from its ability to produce mathematically correct bounds on ranges, even when floating point arithmetic is used. This is achieved with *directed roundings*: The IEEE arithmetic standard [25] for floating point arithmetic, presently almost universally adopted, specifies that it be possible to round the result of a floating point operation in one of four ways, two of which are

1. to the nearest floating point number less than the exact result (rounding down), and
2. to the nearest floating point number greater than the exact result (rounding up).

To implement interval arithmetic on a machine, the operations for computing the lower end point of a result interval are consistently rounded down, and the operations for computing the upper end point are consistently rounded up. Thus, the result of a machine interval evaluation of a expression is a machine interval that contains the mathematically exact interval value, which in turn contains the true result. Hence, the computed interval necessarily contains the mathematically correct result.

For example, take, say, $f(x) = x^2 - 9x$ over the interval $\mathbf{x} = [4, 5]$. An interval evaluation gives $f(x) \in [4, 5]^2 - 9[4, 5] = [16, 25] + [-45, -36] = [-29, -11]$. Thus, even though $[-34, -15]$ is an overestimation of the exact range

$$\{f(x), x \in [4, 5]\} = [-20.25, -20] \subseteq [-34, -15],$$

the computation constitutes a mathematical proof that there are no zeros of $f$ in $[4, 5]$. Furthermore, the interval arithmetic can be carried out with the same steps as floating point arithmetic, and hence range bounds can be obtained for many functions.

In particular, machine implementations of interval arithmetic have interval versions of standard functions such as $\sin(x)$ and $e^x$. These are implemented by considering monotonicity properties of these functions, with careful mathematical error estimation, and with outward rounding, so that, e.g., $\sin(\mathbf{x})$ contains the actual range of sin over $\mathbf{x}$. For details, see [14, 15], or the other references cited above.

The book [19] is a survey of computational complexity results related to bounding the range of functions with interval arithmetic and related to other problems associated with interval computations.

1.1.2. *Notation*

Throughout, we denote scalars and vectors by lower case, and we denote matrices by upper case. Intervals, interval vectors (also called "boxes"), and interval matrices are denoted by boldface. For instance, $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ denotes an interval vector, $A = (a_{i,j})$ denotes a point matrix, and $\mathbf{A} = (\mathbf{a}_{i,j})$ denotes an interval matrix. Real $n$-space is denoted by $\mathbb{R}^n$. Similarly, complex $n$-space is denoted by $\mathbb{C}^n$. The set of $n$-dimensional real interval vectors (also known is "boxes") is denoted by $\mathbb{IR}^n$, while the set of $n$-dimensional complex interval vectors is denoted by $\mathbb{IC}^n$.

The *midpoint* of an interval or interval vector $\mathbf{x} = [\underline{x}, \bar{x}]$ is denoted by $n(\mathbf{x}) = (\underline{x} + \bar{x})/2$. The *width* of $\mathbf{x}$ is denoted by $W(\mathbf{x}) = \bar{x} - \underline{x}$. Although the choice of norm is not critical in the analysis, it is convenient to interpret $\|x\|$ in our discussions below to be the infinity norm of the vector $x$. The boundary of the box $\mathbf{x}$, consisting of $2n$ $(n-1)$-dimensional boxes, will be denoted by $\partial\mathbf{x}$.

A vector-valued function $F\colon \mathbf{x} \subset \mathbb{R}^n \to \mathbb{R}^n$ is denoted by

$$F(x) = (f_1(x), f_2(x), \ldots, f_n(x)). \tag{2}$$

The $d$th order partial derivative of $f_i$ at $x$ with respect to variables $x_{k_1}$, $x_{k_2}, \ldots, x_{k_d}$ (where some of the indices $k_i$ may be repeated) is denoted by

$$\frac{\partial^d f_n}{\partial x_{k_1} \ldots \partial x_{k_d}}(x).$$

We denote the Jacobi matrix of $F$ at $x$ by $F'(x)$, and we denote its determinant by

$$|F'(x)| = \left| \frac{\partial F}{\partial x_1 \ldots x_n}(x) \right|.$$

We have occasion to consider extensions of $F$ into complex space. We identify a complex vector $z = (z_1, z_2, \ldots, z_n) \in \mathbb{C}^n$ with the vector

$(x, y) = (x_1, y_1, x_2, y_2, \ldots, x_n, y_n) \in \mathbb{R}^{2n}$ with $z_k = x_k + iy_k$, where $i$ is the imaginary unit, and we identify real vectors $x = (x_1, \ldots, x_n)$ with $x_1, 0, \ldots, x_n, 0) \in \mathbb{R}^{2n}$. Similarly, if $F$ is as in (2), we write $f$ in terms of its real and imaginary components as

$$f_k(z) = u_k(x, y) + iv_k(x, y), \qquad 1 \leqslant k \leqslant n. \tag{3}$$

If $\mathbf{z} \in \mathbb{IR}^n$ and we identify $\mathbf{z}$ with $(\mathbf{x}, \mathbf{y}) \in \mathbb{IR}^{2n}$ as above, then we write

$$\mathbf{z} = (\mathbf{x}, \mathbf{y}) = (\mathbf{x}_1, \mathbf{y}_1, \mathbf{x}_2, \mathbf{y}_2, \ldots, \mathbf{x}_n, \mathbf{y}_n).$$

Geometrically, $\mathbf{z}$ is a rectangular parallelepiped with $4n$ $(2n-1)$-dimensional faces. We denote the faces by

$$x_{\underline{k}} \equiv (\mathbf{x}_1, \mathbf{y}_1, \ldots, \mathbf{x}_{k-1}, \mathbf{y}_{k-1}, \underline{x}_k, \mathbf{y}_k, \mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \ldots, \mathbf{x}_n, \mathbf{y}_n),$$
$$x_{\bar{k}} \equiv (\mathbf{x}_1, \mathbf{y}_1, \ldots, \mathbf{x}_{k-1}, \mathbf{y}_{k-1}, \bar{x}_k, \mathbf{y}_k, \mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \ldots, \mathbf{x}_n, \mathbf{y}_n),$$
$$y_{\underline{k}} \equiv (\mathbf{x}_1, \mathbf{y}_1, \ldots, \mathbf{x}_{k-1}, \mathbf{y}_{k-1}, x_k, \underline{y}_k, \mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \ldots, \mathbf{x}_n, \mathbf{y}_n),$$

and

$$y_{\bar{k}} \equiv (\mathbf{x}_1, \mathbf{y}_1, \ldots, \mathbf{x}_{k-1}, \mathbf{y}_{k-1}, x_k, \bar{y}_k, \mathbf{x}_{k+1}, \mathbf{y}_{k+1}, \ldots, \mathbf{x}_n, \mathbf{y}_n).$$

With this, define $\tilde{F}: \tilde{\mathbf{D}} \subset \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ by $\tilde{F} = (u_1, v_1, \ldots, u_n, v_n)$, and define

$$\tilde{F}_{\neg u_n}(x, y) \equiv (u_1(x, y), v_1(x, y), \ldots, u_{n-1}(x, y), v_{n-1}(x, y), v_n(x, y)). \tag{4}$$

We will use $\tilde{F}_{\neg u_n}$ below.

### 1.2. *Traditional Computational Fixed Point Theorems*

Suppose $x^*$ satisfies $F(x^*) = 0$, where $F: \mathbf{D} \subset \mathbb{R}^n \to \mathbb{R}^n$, suppose the Jacobi matrix $F'$ is continuous and nonsingular in a neighborhood of $x^*$, suppose that $\mathbf{x} \subseteq \mathbf{D}$ is a small box centered at an approximate solution $\check{x}$ near $x^*$, but $\mathbf{x}$ is sufficiently large to ensure that $x^*$ is also relatively near the center of $\mathbf{x}$. For a precise analysis of when $\mathbf{x}$, $\check{x}$, and $x^*$ have the proper relationship to each other, see the analysis in [15, Sect. 6.2.2], along with [15, Theorem 1.19, p. 62] (originally [20, Theorem 5.1.7] and earlier).

In general, the condition that $x^*$ be near the center of $\mathbf{x}$ cannot be arranged with certainty, since $\check{x}$ has been obtained with some floating-point algorithm that uses a heuristic stopping tolerance. The subsequent verification step, involving methods introduced here, will succeed in proving a

unique solution if this condition is satisfied, and will fail if the condition is not satisfied. When we speak of the number of operations required to complete such a verification algorithm, we will be speaking of the maximum number of operations the algorithm will take before it either verifies the solution or fails. In other words, such algorithms have three possible outputs:

- "a unique solution in **x** has been proven,"
- "no solution in **x** has been proven,"
- "this algorithm cannot prove either a unique solution or no solution in **x**."

Classes of functions and algorithms for which the third possibility (that is, the lack-of-proof possibility) can always be avoided are presently unknown; some negative results are proven in [22]. We also speculate that, for $F$ whose components are Lipschitz, if we assume something about the accuracy of $\check{x}$ as an approximation to $x^*$, about the choice of **x**, and about the error in the second-order Taylor polynomial for the components of $F$ about $x^*$, then the bounding techniques used in [22, Chap. 5] may be used to show that the algorithms described below and in the references will always complete in $\mathcal{O}(n^3)$ total operations (arithmetic operations and function evaluations) with the result "a unique solution in **x** has been proven."

Since traditional computational fixed point theorems, based on interval Newton methods, are explained in [15, 20] and the other references in Section 1.1.1 above, we give only a skeletal outline here. (In the interest of clear exposition, we also do not state results in their most general form.)

As explained in the references, the automatic theorem-proving properties of interval Newton methods are based on combining fundamental calculus concepts such as the mean value theorem, the range-inclusion property of interval arithmetic, and classical fixed point theorems such as the Brouwer fixed point theorem or Miranda's Theorem. The following theorems give examples of such results.

DEFINITION 1. An interval-valued matrix **A** is a Lipschitz matrix for $F$ over **x** provided, for every $x \in \mathbf{x}$ and $\tilde{x} \in \mathbf{x}$, there is an $A \in \mathbf{A}$ such that

$$F(x) - F(\tilde{x}) = A(x - \tilde{x}).$$

For example, any matrix obtained by computing the entries of the Jacobi matrix over **x** with interval arithmetic is a Lipschitz matrix for $F$ over **x**.

An interval Newton method is defined by an iteration of the form

$$\tilde{\mathbf{x}} = \mathbf{N}(F; \mathbf{x}, \check{x}) = \check{x} + \mathbf{v}, \tag{5}$$

where

$$\Sigma(\mathbf{A}, -F(\check{x})) \subset \mathbf{v}, \tag{6}$$

where $\mathbf{A}$ is a Lipschitz matrix for $F$ over $\mathbf{x}$ and where

$$\Sigma(\mathbf{A}, -F(\check{x})) = \{x \in \mathbb{R}^n \mid \exists A \in \mathbf{A} \text{ with } Ax = -F(\check{x})\}.$$

Here $\check{x}$ is some point in $\mathbf{x}$ (often taken to be its midpoint) that, in the context of this paper, we consider to be an approximate solution.

THEOREM 1 ([15, Theorem 1.19, p. 62], originally from [20]). *Suppose* $\tilde{x} = \mathbf{N}(F; \mathbf{x}, \check{x})$ *is the image of* $\mathbf{x}$ *and* $\check{x}$ *under an interval Newton method. If* $\tilde{x} \subseteq \mathbf{x}$, *it follows that there exists a unique solution of* $F(x) = 0$ *within* $\mathbf{x}$.

Various methods, related to but not exactly the same as common floating-points methods, are used to compute the interval vector $\mathbf{v}$ bounding the solution set $\Sigma(\mathbf{A}, -F(\check{x}))$. For instance, the *preconditioned interval Gauss–Seidel method* has various attractive properties for this purpose.

DEFINITION 2. The preconditioned interval Gauss–Seidel image $\mathbf{GS}(F; \mathbf{x}, \check{x})$ of a box $\mathbf{x}$ is defined as $\mathbf{GS}(F; \mathbf{x}, \check{x}) \equiv (\tilde{x}_1, \ldots, \tilde{x}_n)$, where $\tilde{\mathbf{x}}_i$ is defined sequentially for $i = 1$ to $n$ by

$$\tilde{\mathbf{x}}_i \equiv x_i \cap (\check{x}_i - N_i/(Y_i\mathbf{A}_i)),$$

where

$$\mathbf{N}_i = Y_iF(\check{x}) + \sum_{j=1}^{i-1} Y_i\mathbf{A}_j(\tilde{\mathbf{x}}_j - \check{x}_j) + \sum_{j=i+1}^{n} Y_i\mathbf{A}_j(\mathbf{x}_j - \tilde{x}_j),$$

and where $\check{x} = (\check{x}_1, \ldots, \check{x}_n)^T$ is an initial guess point, $Y\mathbf{A} \in \mathbb{IR}^{n\times n}$ and $YF(\check{x})$ are the matrix and right-hand-side vector for the preconditioned interval system $Y\mathbf{A}(x - \check{x}) = -YF(\check{x})$, $Y \in \mathbb{R}^{n\times n}$ is a point preconditioning matrix, $Y_i$ denotes the $i$th row of $Y$, and $\mathbf{A}_j$ denotes the $j$th column of $\mathbf{A}$.

It is not hard to show that, for (6) to hold, $\mathbf{A}$ cannot contain a singular matrix. However, as mentioned above, under natural conditions on the interval extension $\mathbf{A}$ of the Jacobi matrix, assuming the Jacobi matrix $F'(x^*)$ is nonsingular, assuming a box $\mathbf{x}$ can be constructed about an approximate solution $\check{x}$ that is small enough for a quadratic model of $F$ to be accurate but large enough for $x^*$ (approximated by $\check{x}$) to be near the middle, and assuming the preconditioner matrix (the point matrix $Y$ in Definition 2 above) transforms the interval extension $\mathbf{F}'(\mathbf{x}) = \mathbf{A}$ into a diagonally dominant form, then one sweep of the preconditioned interval Gauss–Seidel method will result in $\tilde{\mathbf{x}} \subset \mathbf{x}$. The assumptions are general for

$\mathscr{C}^2$ functions, since a box can be constructed about an accurate approximate solution to ensure that they hold; see the analysis in [15, Sects. 1.5 and 6.2.2]. It is clear that the interval Gauss–Seidel method carries out these computations in $\mathcal{O}(n^3)$ operations.

1.2.1. *Background for the Singular Case*

If the Jacobi matrix $F'(x^*)$ is singular, then $\Sigma(\mathbf{A}, -F(\check{x}))$ is unbounded, and, hence, the condition in Theorem 1 cannot hold. For this reason, common thinking has been that (1) cannot be done when $F'(x^*)$ is singular or excessively ill-conditioned. However, as explained in [4, 17] if $\mathbb{R}^n$ in (1) is replaced by $\mathbb{C}^n$, then, in principle, existence and uniqueness can still be verified. The steps of the algorithms for this singular-case verification are outwardly similar to the non-singular case, except that there is an extra low-dimensional search. In [17], we exhibited algorithms for the rank-defect-1 case, i.e., when the null space of $F'(x^*)$ has dimension 1; we showed theoretically that these algorithms complete (either verifying existence and uniqueness, or verifying no solution, or stating that it is unknown, as explained above) in $\mathcal{O}(n^3)$ operations; we illustrated this dependency on dimension with actual computations on a discretization of a model nonlinear eigenvalue problem, with dimension ranging from 2 to 320.

Our algorithms are based on rigorous computation of the topological index $d = \mathrm{d}(\tilde{F}, \mathbf{z}, 0)$ of the complex extension $\tilde{F}$ of the map $F$ (defined in Section 1.1.2 above) over a box $\mathbf{z}$ of appropriate size centered at the approximate solution $\check{x}$. This topological index $d$ gives the number of solutions, to within "multiplicity," of $\tilde{F}(z) = 0$ within $\mathbf{z}$; see [4, 17] for a review and references, and see the next section below for an introductory clarification of the concept of topological index. The algorithm in [17] was specific to the case where $d = 2$, although we subsequently discovered (see [4, 16]) that the algorithm easily generalizes to arbitrary index. Our algorithm for $d > 2$ is also an $\mathcal{O}(n^3)$ algorithm.

The dissertation [4] also contains a theoretical study and an algorithm dealing directly with $F\colon \mathbf{x} \to \mathbb{R}^n$, where $\mathbf{x} \subset \mathbb{R}^n$, rather than dealing with the complex extension $\tilde{F}\colon \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ defined in Section 1.1.2 above. Also, a heuristic in [4, 16, Sect. 5] is effective at guessing the topological index $d$ of the complex extension at a solution $x^*$ with a rank-1 singularity, that is, at a solution for which the rank of $F'(x^*)$ is $(n-1)$. See Section 2 below for an explanation of the topological index and how the rank of $F'(x^*)$ enters the computations. Furthermore, if the topological index $d$ of the complex extension $\tilde{F}$ happens to be odd, then the topological index of $F$ in $\mathbb{R}^n$ must be either 1 or $-1$, and the value of this real degree can be proven with a computational algorithm much more efficiently (but still with $\mathcal{O}(n^3)$ operations) than the corresponding verification in the complex extension. (See

[4, Sect. 3, 5].) This real-space verification does not guarantee a unique solution or guarantee the multiplicity of the solution, as in the complex space verification of $d$, but it does guarantee existence. However, the real-space verification has the additional theoretical advantage that an actual solution to $F(x) = 0$ has been verified to exist within $\mathbf{x} \subset \mathbb{R}^n$, while the complex computations only verify that a solution, possibly with imaginary components, exists within a small region in complex space containing $\mathbf{x} \subset \mathbb{R}^n$.

While we have implemented algorithms and completed experiments for verifying the topological index $d$ of $\tilde{F}$ when $d \geqslant 2$ and the dimension of the null space of $F'(x^*)$ is one (ibid.), we are presently working on the algorithms for verifying odd topological indices $\pm 1 = (\mathrm{d}(F, \mathbf{x}, 0)$ of real $F$: $\mathbb{R}^n \to \mathbb{R}^n$; we expect this to be straightforward and successful. However, interesting development remains for the case where the dimension of the null space of $F(x^*)$ is greater than one. Here, we present this case, pointing out opportunities and difficulties. In Section 2, we explain our general framework for computing the topological index, while we discuss the higher-dimensional null-space case in Section 3. We discuss the relationship to other methods of handling similar solutions in Section 4, and we speculate on the eventual usefulness and limitations of our techniques in Section 5.

## 2. TOPOLOGICAL INDEX COMPUTATIONS: THE GENERAL SETTING AND THE RANK 1 DEFECT CASE

Our computations are based on

1. preconditioning the system $F(x) = 0$ by multiplying by a constant matrix $Y$ so the Jacobi matrix for $YF(x^*)$ is approximately diagonal, except in $p$ rows, where the dimension of the null space of $F'(x^*)$ is $p$;

2. constructing a box $\mathbf{x}$, with astutely chosen coordinate widths, centered at the approximate solution $\tilde{x}$;

3. computing the Brouwer degree of $YF$, and hence of $F$, over $\mathbf{x}$ searching the $(n-1)$-dimensional sides of $\mathbf{x}$ to verify solutions of a certain system of equations derived from the components of $YF$.

Details of these ideas, as well as a review of properties of the Brouwer degree and references to comprehensive introductions, appear in [4, 17]. Of interest here is the fact that, because of the form of the preconditioned system, the search on the boundary can be greatly streamlined. In particular, previous general algorithms for the topological degree, such as the heuristic algorithms in [12, 24], as well as the rigorous algorithm in [1],

have running times that depend exponentially on $n$. In fact, as shown in [22], a lower bound on the complexity of the degree for Lipschitz functions is exponential, and an algorithm is given in [22].

In contrast, in the preconditioned system $YF$, one can, in effect, express $(n-p)$ of the variables in terms of $p$ variables, where $p$ is the dimension of the null space of the Jacobi matrix $F'(x^*)$. When $p=1$, if the box dimensions are chosen appropriately, all but four of the $4n$ sides of the box in $\mathbb{C}^n$ (treated as a box in $\mathbb{R}^{2n}$) may be eliminated with simple interval evaluations, and the remaining four $(n-1)$-dimensional sides may be handled with one-dimensional searches. Easily-obtainable approximations to the solutions of the system derived from $YF$ further facilitate these one-dimensional searches. Here, we present those details of that process relevant to studying generalization to $p>1$.

Following [4, 17] we observe that, if the rank defect of $F'(x^*)$ is $p$, then the preconditioner $Y$ can be formed as one would compute an inverse of $F'(x^*)$, except with an incomplete LU-factorization based on full pivoting. (Also, $x^*$ is unknown in practice, and we actually compute the preconditioner based on the matrix formed from the midpoints of the elements of the interval extension $F'(\mathbf{x})$ of the Jacobi matrix.) The resulting preconditioned Jacobi matrix, to within a column permutation, has the form in Fig. 1. Hence, if we assume $F(z)=(f_1(z_1, \ldots, z_n), \ldots, f_n(z_1, \ldots, z_n))$ has already been so preconditioned, then, for the rank-1 defect case $p=1$, the components of $f$ have the form

$$f_k(z)=(z_k-x_k^*)+\frac{\partial f_k}{\partial z_n}(x^*)(z_n-x_n^*)+\mathcal{O}(\|z-x^*\|^2) \qquad \text{for} \quad 1\leqslant k\leqslant n-1, \tag{7}$$

$$\begin{aligned} f_n(z)=&\frac{1}{2!}\sum_{k_1=1}^{n}\sum_{k_2=1}^{n}\frac{\partial^2 f_n}{\partial x_{k_1}\,\partial x_{k_2}}(x^*)(z_{k_1}-x_{k_1}^*)(z_{k_2}-x_{k_2}^*)+\cdots \\ &+\frac{1}{d!}\sum_{k_1=1}^{n}\cdots\sum_{k_d=1}^{n}\frac{\partial^d f_n}{\partial x_{k_1}\ldots\partial x_{k_d}}(x^*)(z_{k_1}-x_{k_1}^*)\cdots(z_{k_d}-x_{k_d}^*) \\ &+\mathcal{O}(\|z-x^*\|^{d+1}), \end{aligned} \tag{8}$$

for some $d\geqslant 2$. (See [4, 16].) With the notation from Section 1.1.2, we may translate the above forms for $f_k$ and $f_n$ into the complex setting $\tilde{f}_k(z)=\tilde{f}_k(x, y)=u_k(x, y)+iv_k(x, y)$ to obtain

$$u_k(x, y)=(x_k-x_k^*)+\frac{\partial f_k}{\partial x_n}(x^*)(x_n-x_n^*)+\mathcal{O}(\|(x-x^*, y)\|^2), \tag{9}$$

$$v_k(x, y)=y_k+\frac{\partial f_k}{\partial x_n}(x^*)\,y_n+\mathcal{O}(\|x-x^*, y)\|^2), \tag{10}$$

$$YF'(x^*) = \begin{pmatrix} & & & & \overbrace{\quad}^{p} \\ 1 & 0 & \dots & 0 & *\dots* \\ 0 & 1 & 0\dots & 0 & *\dots* \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 1 & *\dots* \\ 0 & \dots & 0 & 0 & 0\dots 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & 0\dots 0 \end{pmatrix}$$

**FIG. 1.** Derivative of a singular system of rank $n-p$ preconditioned with an incomplete *LU* factorization, where "$*$" represents a non-zero element.

We will compute $\mathrm{d}(\tilde{F}, \mathbf{z}, 0)$, where $\mathbf{z} = (\mathbf{x}, \mathbf{y})$ is chosen according to (11) and (12) below. To compute the degree $\mathrm{d}(\tilde{F}, \mathbf{z}, 0)$ we will consider $\tilde{F}_{\neg u_n}$ (see Section 1.1.2 above) on the boundary of $\mathbf{z}$. The boundary of $\mathbf{z}$ consists of the $4n$ faces $\mathbf{x}_{\underline{1}}$, $\mathbf{x}_{\bar{1}}$, $\mathbf{y}_{\underline{1}}$, $\mathbf{y}_{\bar{1}}$, ..., $\mathbf{x}_{\underline{n}}$, $\mathbf{x}_{\bar{n}}$, $\mathbf{y}_{\underline{n}}$, $\mathbf{y}_{\bar{n}}$. The box coordinates $\mathbf{x}_k$, $\mathbf{y}_k$, $1 \leqslant k \leqslant n$ are chosen so $\mathbf{x}_k$ is centered on $\check{x}_k$ and $\mathbf{y}_k$ is centered on 0, $1 \leqslant k \leqslant n$, and so the widths $\mathrm{w}(\mathbf{x}_n)$ and $\mathrm{w}(\mathbf{y}_n)$ obey

$$\mathrm{w}(\mathbf{x}_n) \leqslant \frac{1}{2} \min_{1 \leqslant k \leqslant n-1} \left\{ \frac{\mathrm{w}(\mathbf{x}_k)}{|\partial f_k / \partial x_n(\check{x})|} \right\}, \tag{11}$$

$$\mathrm{w}(\mathbf{y}_n) \leqslant \frac{1}{2} \min_{1 \leqslant k \leqslant n-1} \left\{ \frac{\mathrm{w}(\mathbf{y}_k)}{|\partial f_k / \partial x_n(\check{x})|} \right\}. \tag{12}$$

The motivation for this choice of widths is to avoid solutions of $u_k = 0$ on $\mathbf{x}_{\underline{k}}$ and $x_{\bar{k}}$, $1 \leqslant k \leqslant n-1$, and to avoid solutions of $v_k = 0$ on $\mathbf{y}_{\underline{k}}$ and $\mathbf{y}_{\bar{k}}$, $1 \leqslant k \leqslant n-1$. In particular, assuming $\mathcal{O}(\|x-x^*, y)\|^2)$ is zero in (9) and solving for $x_n - x_n^*$ gives

$$x_n - x_n^* = -\frac{x_k - x_k^*}{\partial f_k / \partial x_n(x^*)}.$$

Taking widths of both sides gives

$$\mathrm{w}(x_n) = \frac{\mathrm{w}(x_k)}{|\partial f_k / \partial x_n(x^*)|},$$

a condition that must hold if $u_k = 0$. Thus, if $\mathcal{O}(\|x-x^*, y)\|^2)$ is zero and

$$\mathrm{w}(x_n) < \frac{\mathrm{w}(x_k)}{|\partial f_k / \partial x_n(x^*)|},$$

then there can be no solutions of $u_k = 0$ on either $x_{\underline{k}}$ or $x_{\bar{k}}$. If we assume that $\mathcal{O}(\|x-x^*, y)\|^2)$ is non-zero but $x$ is sufficiently close to $x^*$ to ensure $\mathrm{w}(\mathcal{O}(\|(x-x^*, y)\|^2)) < \mathrm{w}(x_k)$, then, because intervals $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ always obey $\mathrm{w}(\boldsymbol{\alpha}+\boldsymbol{\beta}) = \mathrm{w}(\boldsymbol{\alpha})+\mathrm{w}(\boldsymbol{\beta})$,

$$\mathrm{w}(x_n) < \frac{1}{2}\frac{\mathrm{w}(x_k)}{|\partial f_k/\partial x_n(x^*)|}$$

still ensures that $u_k \neq 0$ on $x_{\underline{k}}$ and $x_{\bar{k}}$. Requiring this condition to hold for all $k$, $1 \leqslant k \leqslant n$ gives (11). A similar reasoning, applied to $y_k$, $y_n$ and $v_k$ in lieu of $x_k$, $x_n$ and $u_k$ gives (12).

We verify the value of the topological degree from a formula composed of the algebraic signs of determinants of Jacobi matrix values of $\tilde{F}_{\neg u_n}$. Specifically, provided $\tilde{F}_{\neg u_n}(x, y) \neq 0$ for $(x, y) \in \mathbf{x}_{\underline{k}}, \mathbf{x}_{\bar{k}}, \mathbf{y}_{\underline{k}}, \mathbf{y}_{\bar{k}}$, $1 \leqslant k \leqslant n-1$, the degree $\mathrm{d}(\tilde{F}, \mathbf{z}, 0)$ may be computed with the formula

$$\begin{aligned}
\mathrm{d}(\tilde{F}, \mathbf{z}, 0) = &- \sum_{\substack{x_n = \underline{x}_n \\ \tilde{F}_{\neg u_n}(x, y) = 0 \\ u_n(x, y) > 0}} \operatorname{sgn}\left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1\, y_1 \cdots x_{n-1}\, y_{n-1}\, y_n}(x, y)\right| \\
&+ \sum_{\substack{x_n = \bar{x}_n \\ \tilde{F}_{\neg u_n}(x, y) = 0 \\ u_n(x, y) > 0}} \operatorname{sgn}\left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1\, y_1 \cdots x_{n-1}\, y_{n-1}\, y_n}(x, y)\right| \\
&+ \sum_{\substack{y_n = \underline{y}_n \\ \tilde{F}_{\neg u_n}(x, y) = 0 \\ u_n(x, y) > 0}} \operatorname{sgn}\left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1\, y_1 \cdots x_{n-1}\, y_{n-1}\, y_n}(x, y)\right| \\
&- \sum_{\substack{y_n = \bar{y}_n \\ \tilde{F}_{\neg u_n}(x, y) = 0 \\ u_n(x, y) > 0}} \operatorname{sgn}\left|\frac{\partial \tilde{F}_{\neg u_n}}{\partial x_1\, y_1 \cdots x_{n-1}\, y_{n-1}\, y_n}(x, y)\right|. \qquad (13)
\end{aligned}$$

(Using the development in [24, Sect. 4.2], we derived this formula as Theorem 5.1 of [17]. The formula ultimately results from the recursive nature of the topological degree, that is, the formula is an expression of the property that the topological degree can be written in terms of the topological indices of lower-dimensional maps with respect to a selected point on the boundary of the original region.)

If the coordinate extents are chosen as in (11), then, as explained below (9) and (10), (9) forces $u_k \neq 0$ on $\mathbf{x}_{\underline{k}}$ and $\mathbf{x}_{\bar{k}}$, $1 \leqslant k \leqslant n-1$ when $\mathcal{O}(\|(x-x^*, y)\|^2)$ is small; similarly, if the coordinate extents are chosen as in (12), then (10) forces $v_k \neq 0$ on $\mathbf{y}_{\underline{k}}$ and $\mathbf{y}_{\bar{k}}$, $1 \leqslant k \leqslant n-1$ for $\mathcal{O}(\|(x-x^*, y)\|^2)$ small. However, since we don't compute the value

$\mathcal{O}(\|(x-x^*, y)\|^2)$, $u_k \neq 0$ and $v_k \neq 0$ must be verified. Our verification algorithms, such as our algorithm in Section 6 of [17] or Algorithm 1 of [16], verify $u_k \neq 0$ and $v_k \neq 0$ by computing the interval values $\mathbf{u}(\mathbf{x}_{\underline{k}})$, $\mathbf{u}(\mathbf{x}_{\bar{k}})$, $\mathbf{v}(\mathbf{y}_{\underline{k}})$, and $\mathbf{v}(\mathbf{y}_{\bar{k}})$. Formula (13) is then used by systematic search (made rigorous with interval computations) of the four faces $\mathbf{x}_{\underline{n}}$, $\mathbf{x}_{\bar{n}}$, $\mathbf{y}_{\underline{n}}$, and $\mathbf{y}_{\bar{n}}$ for solutions to $\tilde{F}_{\neg u_n} = 0$. The search is reduced to a one-dimensional search over the $y_n$ coordinate on $\mathbf{x}_{\underline{n}}$ and $\mathbf{x}_{\bar{n}}$ and a one-dimensional search over the $x_n$ coordinate on $\mathbf{y}_{\underline{n}}$ and $\mathbf{y}_{\bar{n}}$ by formally solving the rigorous interval enclosure for $v_k$ corresponding to (12) for $\mathbf{y}_k$ in terms of $\mathbf{y}_n$ and formally solving the rigorous interval enclosure for $u_k$ corresponding to (11) for $\mathbf{x}_k$ in terms of $\mathbf{x}_n$.

For example, suppose the task is to search $\mathbf{x}_{\underline{n}}$ for solutions to $\tilde{F}_{\neg u_n}(x, y) = 0$. Then $x_n$ is fixed at $\underline{x}_n$, and we may use mean-value interval extensions of $u_k$, $1 \leqslant k \leqslant n-1$ corresponding to (9):

$$u_k(x, y) \in u_k(\check{x}, 0) + \sum_{j=1}^{n} \frac{\partial \mathbf{u}_k}{\partial \mathbf{x}_j}(\mathbf{x}, \mathbf{y})(\mathbf{x}_j - \check{x}_j) + \sum_{j=1}^{n} \frac{\partial \mathbf{u}_k}{\partial \mathbf{y}_j}(\mathbf{x}, \mathbf{y})\, \mathbf{y}_j, \qquad (14)$$

Because we are assuming preconditioning, as in Fig. 1, that puts $u_k$ in the form (9), all partial derivatives in the sums in (14) have small absolute values except

$$\frac{\partial \mathbf{u}_k}{\partial \mathbf{x}_k}(\mathbf{x}, \mathbf{y}) \qquad \text{and} \qquad \frac{\partial \mathbf{u}_n}{\partial \mathbf{x}_n}(\mathbf{x}, \mathbf{y}).$$

Thus, if $x_n$ and $y_n$ are known, we may formally solve $u_k(x, y) = 0$ in (14) to obtain sharper bounds on $x_k$, $1 \leqslant k \leqslant n-1$. In particular, we compute

$$\tilde{\mathbf{x}}_k = \check{x}_k - \frac{u_k(\check{x}, 0) + \sum_{\substack{j=1 \\ j \neq k}}^{n} \frac{\partial \mathbf{u}_k}{\partial \mathbf{x}_j}(\mathbf{x}, \mathbf{y})(\mathbf{x}_j - \check{x}_j) + \sum_{j=1}^{n} \frac{\partial \mathbf{u}_k}{\partial \mathbf{y}_j}(\mathbf{x}, \mathbf{y})\, \mathbf{y}_j}{\frac{\partial \mathbf{u}_k}{\partial \mathbf{x}_k}(\mathbf{x}, \mathbf{y})} \qquad (15)$$

with interval arithmetic, to obtain $\mathrm{w}(\tilde{\mathbf{x}}_k) \ll \mathrm{w}(\mathbf{x}_k)$, such that any solutions of $\tilde{F}_{\neg u_n}(x, y) = 0$ must have $x_k \in \tilde{\mathbf{x}}_k$. (Here, $\check{x}$ is computed to hopefully be close to the actual solution $x^*$, $F(x^*) = 0$. Rigor is not lost if this is not so, since the interval evaluation of the right member of (15) is always valid in the sense that an incorrect conclusion will never be reached, regardless of whether or not it gives $\mathrm{w}(\tilde{\mathbf{x}}_k)$ small. That is mathematical correctness is never lost, only the ability to verify, when the assumptions are not valid.)

On $\mathbf{x}_{\underline{n}}$, $\mathbf{x}_n$ is fixed at $\underline{x}_n$, so (15) will likely give narrow bounds $\tilde{\mathbf{x}}_k$ on $x_k$, $1 \leqslant k \leqslant n-1$. If we also knew $y_n$, we could similarly use $v_k(x, y) = 0$ and a mean value extension for $v_k$ to compute tight bounds on $y_k$, $1 \leqslant k \leqslant n-1$. Since we do not know $y_n$ on $\mathbf{x}_{\underline{n}}$ we treat $y_n$ as a variable, and define a function

$$g(y_n) = v_n(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}),$$

where $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_{n-1}, \underline{x}_n)$ and $\tilde{\mathbf{y}} = (\tilde{y}_1, \ldots, \tilde{y}_{n-1}, y_n)$. We may then systematically search $\mathbf{y}_n = [\underline{y}_n, \bar{y}_n]$, say, by adaptively subdividing $\mathbf{y}_n$ into small intervals. Interval evaluations of $g$ are then used to reject portions of $\mathbf{y}_n$ upon which $g \neq 0$, and a univariate interval Newton method is used to prove existence and uniqueness of solutions of $g = 0$, and hence of $\tilde{F}_{\neg u_n} = 0$ on $\mathbf{x}_{\underline{n}}$. (The cost of this adaptive univariate search depends on the amount of overestimation of the interval extensions and on the modulus of continuity of the function.) Interval arithmetic (with interval Gaussian elimination) is then used to compute the determinants in (13), to finish the mathematically rigorous computation of the first sum in (13). Computations over $\mathbf{x}_{\bar{n}}$, $\mathbf{y}_{\underline{n}}$, and $\mathbf{y}_{\bar{n}}$ proceed similarly. For a complete description of successful search techniques and for further details, see [4, 16, 17].

The one-dimensional search is facilitated with accurate a priori approximations to the solutions of $\tilde{F}_{\neg u_n}(x, y) = 0$ on $\mathbf{x}_{\underline{n}}$, $\mathbf{x}_{\bar{n}}$, and $\mathbf{y}_{\underline{n}}$, and $\mathbf{y}_{\bar{n}}$. Denote

$$\begin{aligned}
&\alpha_k = \frac{\partial f_k}{\partial x_n}(\check{x}), \qquad 1 \leqslant k \leqslant n-1, \\
&\alpha_n = -1, \\
&\Delta_1 = \left| \frac{\partial F}{\partial x_1 \ldots \partial x_n}(\check{x}) \right|. \\
&\Delta_d = \sum_{k_1=1}^{n} \cdots \sum_{k_d=1}^{n} \frac{\partial^d f_n}{\partial x_{k_1} \ldots \partial x_{k_d}}(\check{x})\, \alpha_{k_1} \ldots \alpha_{k_d}, \qquad 2 \leqslant d.
\end{aligned} \tag{16}$$

Then, consider $f_n(z) = (u_n(x, y), v_n(x, y))$, assume the formulas (7) and (8) to be exact when we remove the $\mathcal{O}(\|z - x^*\|^2)$ and $\mathcal{O}(\|z - x^*\|^{d+1})$, and assume the actual topological index is $d$. When the topological index is $d$, it is not hard to show (see [16, Theorem 3.1]) that all terms in (8) except terms of order $d$ vanish. Further assuming $f_k(z) = 0$, $1 \leqslant k \leqslant n-1$, solving for $z_k$ in terms of $z_n$ in (7), and plugging into (8) finally gives

$$f_n(z) = \frac{(-1)^d\, t^d\, \Delta_d}{d!} (z_n - x_n^*)^d. \tag{17}$$

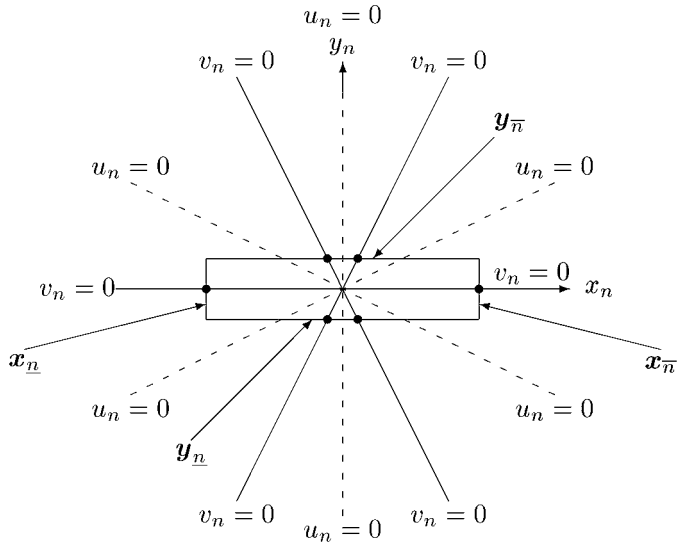


**FIG. 2.** Approximate solutions of $u_n = 0$ and $v_n = 0$ on $\mathbf{x}_{\underline{n}}$, $\mathbf{x}_{\bar{n}}$, $\mathbf{y}_{\underline{n}}$, and $\mathbf{y}_{\bar{n}}$ for $d = 3$. Here, $v_n = 0$ on solid lines and $u_n = 0$ on dashed lines. The thick dots are the solutions of $\tilde{F}_{\neg u_n}(x, y) = 0$ on $\partial\mathbf{x}$.

It follows from (17) that solutions of $v_n = 0$ are at those complex values $z_n = (x_n, y_n)$ whose argument $\theta$ obeys $\theta^d = \pi/2 + \ell\pi$ for some integer $\ell$. This is illustrated for $d = 4$ in Fig. 2. For details, see [4].

Thus, in the one-dimensional rank defect case, one-dimensional subintervals of four one-dimensional intervals, representing $\mathbf{y}_n$ on $\mathbf{x}_{\underline{n}}$ and $\mathbf{x}_{\bar{n}}$ and $\mathbf{x}_n$ on $\mathbf{y}_{\underline{n}}$ and $\mathbf{y}_{\bar{n}}$, can be constructed about approximate solutions of $\tilde{F}_{\neg u_n}(x, y) = 0$. With a univariate interval Newton method applied to the function $g(y_n)$ (or analog) defined above, these subintervals can be rapidly verified to contain unique solutions of $\tilde{F}_{\neg u_n}(x, y) = 0$; the remainder of the four original one-dimensional intervals can be rapidly but rigorously eliminated with interval evaluations of $g$. Actual algorithms appear in [4], numerical results for $d = 2$ appear in [17], and numerical experiments for $d > 2$ appear in [16]. The case $d = 1$ necessarily must have $\Delta_1 \neq 0$, and hence can be handled with ordinary interval Newton methods as explained in Section 1.2 above.

### 2.1. *Numerical Results*

To illustrate the cubic dependence on the dimension $n$, we reported experimental results in [16, 17]. There, we performed the verification process, constructing $\mathbf{x}$ about $\check{x} = x^*$ as described above, for two variable-dimension examples. We used a Sparc 140 MHz Ultra 1 with Sun Fortran

TABLE I

Numerical Results

| Problem | $n$ | Heuristic degree | Succes | Verified degree | CPU time | Time ratio |
|---|---|---|---|---|---|---|
| Example 1 | 5 | 2 | Yes | 2 | 1.13 | |
| Example 1 | 10 | 2 | Yes | 2 | 5.99 | 5.30 |
| Example 1 | 20 | 2 | Yes | 2 | 38.40 | 6.41 |
| Example 1 | 40 | 2 | Yes | 2 | 273.61 | 7.13 |
| Example 1 | 80 | 2 | Yes | 2 | 2198.14 | 8.03 |
| Example 1 | 160 | 2 | Yes | 2 | 13033.22 | 5.93 |
| Example 2 | 5 | 3 | Yes | 3 | 39.27 | |
| Example 2 | 10 | 3 | Yes | 3 | 10.31 | 0.26 |
| Example 2 | 20 | 3 | Yes | 3 | 74.32 | 7.21 |
| Example 2 | 40 | 3 | Yes | 3 | 481.23 | 6.48 |
| Example 2 | 80 | 3 | Yes | 3 | 3805.06 | 7.91 |
| Example 2 | 160 | 3 | Yes | 3 | 33944.20 | 8.92 |

version 1 and our Fortran 90 interval arithmetic package[1] [13]. We tried the following two examples.

EXAMPLE 1 (Motivated from considerations in [11]). Set $f(x) = h(x, t) = (1-t)(Ax-x^2)-tx$, where $A \in \mathbb{R}^{n\times n}$ is the matrix corresponding to central difference discritization of the boundary value problem $-u'' = 0$, $u(0) = u(1) = 0$ and $x^2 = (x_1^2, \ldots, x_n^2)^T$. $t$ was chosen to be equal to $t_1 = \lambda_1/(1+\lambda_1)$, where $\lambda_1$ is the largest eigenvalue of $A$.

EXAMPLE 2. This example is identical to Example 1, except that we set $f(x) = h(x, t) = (1-t)(Ax-x^3)-tx$.

The results appear in Table I. We observe that the algorithm successfully verified the topological index of the solution in all cases, and that the dependency on $n$ is approximately cubic, as predicted by our analysis. The "heuristic degree" is computed by a very fast algorithm that uses a ratio of function norms to distance from the approximate solution. The CPU time is time in seconds for the verified computations, whereas the time ratio is the ratio of CPU time for the dimension of the corresponding table row to the CPU time for the previous table row. For further details, see [16]. Also see [5] for numerical results for the more efficient algorithm in real space.

[1] The newest version of Sun Fortran has an intrinsic interval data type, which would provide better results on Sun Equipment.

## 3. THE HIGHER RANK-DEFECT CASE

When the dimension $p$ of the null space of $F'(x^*)$ is greater than 1, the forms corresponding to (7) and (8) are

$$f_k(z) = (z_k - x_k^*) + \frac{\partial f_k}{\partial z_n}(x^*)(z_n - x_n^*) + \mathcal{O}(\|z - x^*\|^2)$$
$$\text{for} \quad 1 \leqslant k \leqslant n-p, \tag{18}$$

$$f_q(z) = \frac{1}{2!}\sum_{k_1=1}^{n}\sum_{k_2=1}^{n}\frac{\partial^2 f_q}{\partial x_{k_1}\partial x_{k_2}}(x^*)(z_{k_1} - x_{k_1}^*)(z_{k_2} - x_{k_2}^*) + \cdots$$
$$+\frac{1}{d!}\sum_{k_1=1}^{n}\cdots\sum_{k_d=1}^{n}\frac{\partial^d f_q}{\partial x_{k_1}\ldots\partial x_{k_d}}(x^*)(z_{k_1} - x_{k_1}^*)\ldots(z_{k_d} - x_{k_d}^*)$$
$$+\mathcal{O}(\|z - x^*\|^{d+1}), \qquad \text{for} \quad n-p+1 \leqslant q \leqslant n. \tag{19}$$

In this more general setting, (18) can be used as before to eliminate variables from (19). However, $p$ variables remain, and there are $p$ equations left. In general, this system is an arbitrary system of $p$ homogeneous degree-$d$ equations in $p$ variables; to see this, let $n = p$ and *specify* the *complete original* system by

$$f_q(z) = \frac{1}{d!}\sum_{k_1=1}^{p}\cdots\sum_{k_d=1}^{p}\frac{\partial^d f_q}{\partial x_{k_1}\ldots\partial x_{k_d}}(x^*)(z_{k_1} - x_{k_1}^*)\ldots(z_{k_d} - x_{k_d}^*)$$
$$\text{for} \quad 1 \leqslant q \leqslant p, \tag{20}$$

where the partial derivatives are set arbitrarily, subject only to the condition that corresponding mixed partial derivatives are equal. This implies that, in the analogue of the case when $p = 1$, a $p$-dimensional space must be searched. Furthermore, for approximate starting solutions, instead of a simple formula as for (17) and Fig. 2, all solutions to a general $d$-homogeneous system (that is, a system all of whose terms are of degree $d$) of $p$ equations in $p$ unknowns would need to be found. For higher $p$ and $d$, that could be expensive for a verification step that may be small part of another overall algorithm.

The general formula from which (13) was derived is [17, Theorem 2.5]. In particular, consider $F\colon \mathbb{R}^n \to \mathbb{R}^n$, fix a $p$ between 1 and $n$, let $\underline{K_0(s)}$ denote the subset of the integers $k \in \{1, \ldots, n\}$ such that $F_{\neg p} = 0$ has solutions on $\mathbf{x}_{\underline{k}}$ and $\operatorname{sgn}(f_p) = s$ at these solutions, and let $\overline{K_0(s)}$ denote the subset of the integers $k \in \{1, \ldots, n\}$ such that $F_{\neg p} = 0$ has solutions on $\mathbf{x}_{\bar{k}}$

and $\operatorname{sgn}(f_p) = s$ at these solutions, where $s \in \{-1, +1\}$ and $F_{\neg p} = (f_1, \ldots, f_{p-1}, f_{p+1}, \ldots, f_n)$.

THEOREM 2 (Theorem 2.5 from [17]). *Suppose $F \neq 0$ on $\partial \mathbf{x}$, and suppose there is $p$, $1 \leqslant p \leqslant n$, such that*

(1) *$F_{\neg p} \neq 0$ on $\partial \mathbf{x}_{\underline{k}}$, $k = 1, \ldots, n$;*

(2) *$f_p$ has the same sign at all solutions of $F_{\neg p} = 0$, if there are any, on $\mathbf{x}_{\underline{k}}$ or $\mathbf{x}_{\bar{k}}$, $1 \leqslant k \leqslant n$; and*

(3) *the jacobi matrices of $F_{\neg p}$ are nonsingular at all solutions of $F_{\neg p} = 0$ on $\partial \mathbf{x}$.*

*Then*

$$\mathrm{d}(F, \mathbf{x}, 0) = (-1)^{p-1} s \left\{ \sum_{k \in \underline{K_0(s)}} (-1)^k \sum_{\substack{x \in \mathbf{x}_{\underline{k}} \\ F_{\neg p}(x) = 0}} \operatorname{sgn} \left| \frac{\partial F_{\neg p}}{\partial x_1 x_2 \ldots x_{k-1} x_{k+1} \ldots x_n} (x) \right| \right.$$

$$\left. + \sum_{k \in \overline{K_0(s)}} (-1)^{k+1} \sum_{\substack{x \in \mathbf{x}_{\bar{k}} \\ F_{\neg p}(x) = 0}} \operatorname{sgn} \left| \frac{\partial F_{\neg p}}{\partial x_1 x_2 \ldots x_{k-1} x_{k+1} \ldots x_n} (x) \right| \right\}.$$

A straightforward change of notation in Theorem 2 to our complex setting, selection of $s = 1$, and selection of $p = 2n-1$ (corresponding to the component $u_n$) in that theorem gives the general formula

$$\begin{aligned} \mathrm{d}(F, \mathbf{z}, 0) = \mathrm{d}(\tilde{F}, (\mathbf{x}, \mathbf{y}), 0) \\ = - \sum_{k \in \underline{K_0}} \sum_{\substack{x \in \mathbf{x}_{\underline{k}} \\ F_{\neg u_n}(z) = 0}} \operatorname{sgn}(D_1) \\ + \sum_{k \in \overline{K_0}} \sum_{\substack{x \in \mathbf{x}_{\bar{k}} \\ F_{\neg u_n}(z) = 0}} \operatorname{sgn}(D_1) \\ + \sum_{k \in \underline{K_1}} \sum_{\substack{x \in \mathbf{y}_{\underline{k}} \\ F_{\neg u_n}(z) = 0}} \operatorname{sgn}(D_2) \\ - \sum_{k \in \overline{K_1}} \sum_{\substack{x \in \mathbf{y}_{\bar{k}} \\ F_{\neg u_n}(z) = 0}} \operatorname{sgn}(D_2) \end{aligned} \tag{21}$$

where

$$D_1 = \left| \frac{\partial F_{\neg u_n}}{\partial x_1 y_1 x_2 y_2 \ldots x_{k-1} y_{k-1} y_k x_{k+1} y_{k+1} \ldots x_n y_n} (z) \right|$$

and

$$D_2 = \left| \frac{\partial F_{\neg u_n}}{\partial x_1 y_1 x_2 y_2 \ldots x_{k-1} y_{k-1} x_k x_{k+1} y_{k+1} \ldots x_n y_n} (z) \right|.$$

(Note that now, in contrast to (13), any $u_r$, $n-p+1 \leqslant r \leqslant n$ may be chosen for $F_{\neg u_r}$ (i.e., we are not restricted to $r = n$).)

Now, as in the rank-1 defect case $p = 1$, the box $\mathbf{z}$ can be constructed so, if the higher-order terms are not too large (which can be subsequently verified computationally), $u_k$ is nonzero on $\mathbf{x}_{\underline{k}}$ and $\mathbf{x}_{\bar{k}}$, and $v_k$ is nonzero on $\mathbf{y}_{\underline{k}}$ and $\mathbf{y}_{\bar{k}}$ for $1 \leqslant k \leqslant n-p$. In particular, the forms corresponding to (9) and (10) are

$$u_k(x, y) = (x_k - x_k^*) + \sum_{q=n-p+1}^{n} \frac{\partial f_k}{\partial x_q}(x^*)(x_q - x_q^*) + \mathcal{O}(\|(x - x^*, y)\|^2), \tag{22}$$

$$v_k(x, y) = y_k + \sum_{q=n-p+1}^{n} \frac{\partial f_k}{\partial x_q}(x^*)\, y_q + \mathcal{O}(\|(x - x^*, y)\|^2), \tag{23}$$

from which it follows that conditions corresponding to (11) and (12) are

$$\sum_{q=n-p+1}^{n} \left| \frac{\partial f_k}{\partial x_q}(x^*) \right| \mathrm{w}(\mathbf{x}_q) \leqslant \frac{1}{2} \mathrm{w}(\mathbf{x}_k), \qquad 1 \leqslant k \leqslant n-p, \tag{24}$$

$$\sum_{q=n-p+1}^{n} \left| \frac{\partial f_k}{\partial x_q}(x^*) \right| \mathrm{w}(\mathbf{y}_q) \leqslant \frac{1}{2} \mathrm{w}(\mathbf{y}_k), \qquad 1 \leqslant k \leqslant n-p, \tag{25}$$

Therefore, the $4p$ faces $\mathbf{x}_{\underline{q}}$ and $\mathbf{x}_{\bar{q}}$, and $\mathbf{y}_{\underline{q}}$ for $n-p+1 \leqslant q \leqslant n$ cannot be eliminated from consideration in the sums in (21). In fact, instead of being reduced to (13), (21) can only be reduced to

$$\mathrm{d}(F, \mathbf{z}, 0) = \sum_{q=n-p+1}^{n} \left\{ - \sum_{\substack{x_q = x_q \\ \tilde{F}_{\neg u_n}(x, \bar{y}) = 0 \\ u_r(x, y) > 0}} \mathrm{sgn}(D_1) + \sum_{\substack{x_q = \bar{x}_q \\ \tilde{F}_{\neg u_n}(x, y) = 0 \\ u_r(x, y) > 0}} \mathrm{sgn}(D_1) + \sum_{\substack{y_q = y_q \\ \tilde{F}_{\neg u_n}(x, y) = 0 \\ u_r(x, y) > 0}} \mathrm{sgn}(D_2) + \sum_{\substack{y_q = \bar{y}_q \\ \tilde{F}_{\neg u_n}(x, y) = 0 \\ u_r(x, y) > 0}} \mathrm{sgn}(D_2) \right\}. \tag{26}$$

Thus, instead of four one-dimensional intervals to be searched, $4p$ $2p$-dimensional boxes (corresponding to real and imaginary coordinates of the $p$ variables that could not be eliminated) would need to be searched. All in all, it thus appears that the task increases exponentially with the rank defect $p$, and the complication of implementation jumps substantially from $p = 1$ to $p = 2$.

Experimental results will be forthcoming.

## 4. RELATIONSHIP TO OTHER TECHNIQUES

Important in applications, computation and classification of singular or nearly singular solutions and bifurcation points has been studied extensively in the literature. However, most of this work has been based on *computation of approximate solutions* or *analytic* (rather than computational) *classification of the types of singularities*. For example, Griewank and Reddien [8] propose a bordering method (augmenting the system of equations to make it nonsingular) for allowing the point Newton method to converge quickly to a singular solution with rank defect 1. In [9], Griewank discusses the behavior of Newton's method near singular points, then reviews techniques, including bordering and use of higher-order derivative tensors, for fast convergence near singular points. On the analytical side, Govaerts *et al.* [7] review and unify work connecting higher-order tensors with classification of bifurcation points. This work contrasts with ours in the following ways:

- The goal in Griewank *et al.* is to find approximate singular solutions, whereas the goal in our present work is to verify the existence and character of an approximate solution, once it is known.
- While some techniques (reviewed in [9]) involve actually computing higher-order derivatives, execution of our verification process only involves computation of function values and first-order derivatives, regardless of the rank defect of the Jacobi matrix or the order of the singularity. (Our computations are done at points set off sufficiently from the actual singularity for the system to be numerically non-singular.)
- Griewank *et al.* consider only the rank defect 1 case, whereas the approach in this paper applies more generally.
- Although more general, bordering is used in conjunction with Lyapunov–Schmidt reductions in the techniques developed and reviewed in [6, 7]. Again, however, the goal is to *find*, rather than *verify* bifurcation points. Furthermore, some higher-order directional derivative computations are required.

• Our technique verifies a more fundamental property (the topological degree) of a singularity than a particular type of bifurcation point, etc. Furthermore, we have a simple heuristic [4, Sect. 2.3] to determine the value of the degree before verification.

That said, it appears quite possible to use the bordering techniques from [6, 9] in numerical verification processes. In particular, application of an interval Newton method to the bordered system can prove existence and uniqueness of solutions to the bordered system and, hence, existence and uniqueness of singular points of the type consistent with the structure imposed by the bordering or by the Lyapunov–Schmidt reduction. Application of such an interval Newton method would also require $\mathcal{O}(n^3)$ work, as with our technique (assuming a dense Jacobi matrix).

There may also be deeper connections between developments in the Lyapunov–Schmidt reduction and our technique.

## 5. USES AND LIMITATIONS

Verification of topological indices is potentially useful in automatic theorem proving associated with bifurcation theory and practical bifurcation problems. It also could be useful in branch and bound optimization algorithms as described in [10; 15, Chap. 5], to verify feasibility of a set of constraints that happen to be linearly dependent on isolated parts of the feasible set. Although such linear dependencies appear unlikely (or impossible) from a probabilistic point of view, they do occur in practice. Higher-order rank deficiencies also occur in practice. However, the difficulty of verification appears to increase rapidly with the dimension of the null space, and the implementation becomes significantly more complicated between a one-dimensional and a two-dimensional null space.

In any case, the techniques treated here are in general only applicable to isolated solutions. To see this, note that a condition that the topological degree $\mathrm{d}(F, \mathbf{x}, 0)$ be defined is that there be no solutions to $F(x) = 0$ on the boundary of $x$; if a solution $x^*$ to $F(x) = 0$ is not isolated, then $\mathbf{x}$ in general cannot be so chosen.

An algorithm for exhaustively analyzing the solution sets of polynomial systems that have higher-dimensional solution sets is described in [23]. That algorithm, in its present form, does not claim to rigorously verify existence or uniqueness of the approximate solution sets it finds, the algorithms in [23] have successfully determined all solution manifold components for a variety of systems.

## ACKNOWLEDGMENTS

I thank the referees, and, in particular, the second referee and the editor Professor Sikorski, for the careful attention to the manuscript, that much improved the completeness and clarity of this work.

## REFERENCES

1. O. Aberth, Computation of topological degree using interval arithmetic, and applications, *Math. Comp.* **62** (1994), 171–178.
2. G. Alefeld and J. Herzberger, "Introduction to Interval Computations," Academic Press, New York, 1983.
3. G. Bohlender, Bibliography on enclosure methods and related topics, 1996, `fttp://ftp.iam.uni-karlsruhe.de/pub/documents/literature-list/`.
4. J. Dian, "Existence Verification of Higher Degree Singular Zeros of Nonlinear Systems," Ph.D. thesis, University of Louisiana at Lafayette, 2000.
5. J. Dian and R. B. Kearfott, Existence verification for singular zeros of real nonlinear systems, 2001, `http://interval.louisiana.edu/preprints/degree_real.pdf/`.
6. W. Govaerts, Computation of singularities in large nonlinear systems, *SIAM J. Numer. Anal.* **34** (1997), 867–880.
7. W. Govaerts, Yu. A. Kuznetsov, and B. Sijnave, Numerical methods for the generalized Hopf bifurcation, *SIAM J. Numer. Anal.* **38** (2000), 329–346.
8. A. Griewank and G. W. Reddien, Characterization and computation of generalized turning points, *SIAM J. Numer. Anal.* **21** (1984), 176–185.
9. A. Griewank, On solving nonlinear equations with simple singularities or nearly singular solutions, *SIAM Rev.* **27** (1985), 537–564.
10. E. R. Hansen, "Global Optimization Using Interval Analysis," Dekker, New York, 1992.
11. H. Jürgens, H.-O. Peitgen, and D. Saupe, Topological perturbations in the numerical nonlinear eigenvalue and bifurcation problems, *in* "Analysis and Computation of Fixed Points," pp. 139–181, Academic Press, New York, 1980.
12. R. B. Kearfott, An efficient degree-computation method for a generalized method of bisection, *Numer. Math.* **32** (1979), 109–127.
13. R. B. Kearfott, Algorithm 763: `INTERVAL_ARITHMETIC`: A Fortran 90 module for an interval data type, *ACM Trans. Math. Software* **22** (1996), 385–392.
14. R. B. Kearfott, Interval computations—Introduction, uses, and resources, *Euromath Bull.* **2** (1996), 95–112.
15. R. B. Kearfott, "Rigorous Global Search: Continuous Problems," Kluwer Academic, Dordrecht, 1996.
16. R. B. Kearfott and J. Dian, Existence verification for higher-degree singular zeros of complex nonlinear systems, 2000, `http://interval.louisiana.edu/preprints/degree_cplx.0302.pdf/`.
17. R. B. Kearfott, J. Dian, and A. Neumaier, Existence verification for singular zeros of complex nonlinear systems, *SIAM, J. Numer. Anal.* **38** (2000), 360–379.
18. V. Kreinovich, Interval computations web page, 2001; `http://www.cs.utep.edu/interval-comp/main.html`.
19. V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, "Computational Complexity and Feasibility of Data Processing and Interval Computations," Kluwer Academic, Dordrecht, 1998.

20. A. Neumaier, "Interval Methods for Systems of Equations," Cambridge Univ. Press, Cambridge, U.K., 1990.
21. H. Ratschek and J. Rohne, "New Computer Methods for Global Optimization," Wiley, New York, 1988.
22. K. A. Sikorski, "Optimal Solution of Nonlinear Equations," Oxford Univ. Press, New York, 2001.
23. A. J. Sommese, J. Verschelde, and C. W. Wampler, Numerical decomposition of the solution sets of polynomial systems into irreducible components, *SIAM J. Numer. Anal.* **38** (2000), 2022–2046.
24. F. Stenger, An algorithm for the topological degree of a mapping in $\mathbb{R}^n$, *Numer. Math.* **25** (1976), 23–38.
25. D. Stevenson, chairman, "Floating-Point Working Group, Microprocessor Standards, Subcommittee, IEEE Standard for Binary Floating Point Arithmetic," IEEE/ANSI 754-1985, Technical Report, IEEE, 1985.
26. G. W. Walster *et al.*, Forte Fortran/HPC: interval arithmetic, 2000; `http://www.sun.com/forte/fortran/interval/`.